

## **REMARKS**

The claims have been re-listed with their correct numbering. Claims 21-67 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

### **Section 103(a) Rejection:**

The Office Action rejected claims 21-25, 29-39, 43-45, 47-54 and 58-68 under 35 U.S.C. § 103(a) as being unpatentable over Cook, et al. (U.S. Patent 5,727,950) (hereinafter "Cook") in view of Martinez (U.S. Patent 5,546,521),

Regarding claim 21, Applicant respectfully disagrees with the Examiner and submits that Cook fails to teach maintaining a cache specific to help data for one or more user interface components.

Cook teaches a method for interactive, adaptive, and individualized computer-assisted instruction that provides a virtual agent or tutor adapted to each student (Cook, column 5, lines 12-24). Cook's virtual agent presents educational material and is responsive to the student actions and requests (Cook, column 12, lines 46-51, and column 13, lines 8-27). While Cook does teach that the virtual agent provides help data to the student, Cook teaches that the system is advantageously organized *to wait for student* and timer input events and then to respond appropriately to those events (Cook, column 37, line 66 – column 38, line 2). Cook *fails to teach* maintaining a cache *specific to help data for one or more user interface components*. In contrast, Cook teaches that the files or lessons for computer-aided instruction may be downloaded ahead of time. However, these files are not help data for one or more user interface components, but are "educational content such as instructional units, homework assignments, and testing" materials (Cook, column 10, lines 59-64). Further, the files downloaded by Cook are not maintained in a cache specific to help data and Cook fails to teach maintaining such a cache specific to help data.

As the Examiner states, Cook teaches that “[l]arge files can be downloaded in advance of a student session or the student client can cache read-only data across sessions obviating the need for downloading such files” (Cook, column 16, lines 20-22). However, Cook is clearly referring to the student’s client software downloading lesson files and other educational materials for presentation to the student. Cook does not mention anything regarding a cache specific to help data. Further, the files downloaded by Cook are not presented to the user as help data for one or more user interface components. Instead the download files include “computer-assisted instruction materials which present to students a variety of interactive, adaptive, and self-paced computer-assisted instruction and homework materials in a manner which informs the agent of a student’s progress and performance and which permits the agent to manage or control the materials to the student’s pedagogic characteristics” (Cook, column 6, lines 57-64). Clearly such files are not help data for one or more user interface components. In regard to help data, Cook only teaches that its virtual agent provides help data to the student, and that the system is advantageously organized *to wait for student* and timer input events and then to respond appropriately to those events (Cook, column 37, line 66 – column 38, line 2). Thus, Cook clearly fails to teach anything regarding maintaining a cache specific to help data for one or more user interface components.

Further, Cook fails to disclose anything regarding providing help for user interface components. In contrast, Cook teaches that the virtual agent may provide “meta-cognitive help on ongoing instruction” (Cook, column 5, lines 47-49). For example, the student may make meta-requests to the virtual agent, such as “How am I doing?” to which the agent responds (Cook, column 13, lines 15-24). Cook is teaching that the virtual agent can provide educational help relating to a student’s lessons in subjects such as math or writing. This type of help, as described by Cook, is very different from help data for user interface components. Nowhere does Cook discuss anything regarding a help system that provides help data for one or more user interface components.

The Examiner relies upon Martinez to disclose, because Cook fails to teach, loading the help data for a referenced component into the cache if the help data for the referenced component is not in the cache. However, contrary to the Examiner's assertion, Martinez clearly fails to teach such a system. Martinez teaches a help system including dynamically displaying help text for a user interface objects under the mouse, but fails to teach anything regarding a *cache specific to help data* and additionally fails to teach loading the help data for a referenced component into the cache if the help data for the referenced component is not in the cache. In contrast, Martinez teaches that help data is loaded either from dynamic table 41 or static table 40. Specifically, Martinez teaches that in response to determining that an object is under the mouse pointer, the help system first looks in dynamic table 41 and if it doesn't find any help text in dynamic table 41 for the object, the help system looks in static table 40 "to determine whether a help string should be displayed to the user" (Martinez, column 6, line 64 – column 7, line 2). Additionally, Martinez teaches that certain applications, that he calls aware applications, may update the help data in dynamic table 41 (Martinez, column 5, lines 49-55). However, this updating of dynamic table 41 is proactive, and not reactive to not finding help data for a referenced user interface components. In other words, an aware application may modify the specific help text displayed for a particular user interface element, but Martinez does not teach that help data is loaded into dynamic table 41 if no help data for a user interface component is found in dynamic table 41. Instead, as noted above, Martinez teaches looking in static table 40 and displaying any help data found, but Martinez does not teach loading any help data from static table 40 into dynamic table 41. Thus, neither dynamic table 40, nor static table 40 are caches as understood in the art, but instead are merely string tables from which help data is read and displayed to the user.

The Examiner's cited passages from Martinez describe various disparate functions of Martinez' help system including: referring to a table correlating user interface objects with help data in response to determining that the pointer is over a user interface object (Martinez, column 2, lines 49-54), presenting help data from the help systems even if the underlying application does not provide its own help data (Martinez, column 6, lines 1-9), that dynamic help table 41 resides in a shared memory segment

separate from static help table 40 (Martinez, column 5, lines 54-56), and describing an control application, OverSee 37, that conforms to the Common User Access guidelines for its graphical user interface (Martinez, column 6, lines 42-44). None of these sections mentions anything regarding a cache specific to help data, nor about loading into a cache help data for a user interface component for which no help data is currently in the cache.

Further, Martinez teaches that when the mouse pointer is over an object for which the help system does not already have help information, the help system attempts to find the information associated with the object in help tables 40 and 41. If the help system cannot find help data for the object, the help system “returns to monitoring the current screen location of the mouse pointer.” (Martinez, column 8, lines 49-63). Thus, Martinez fails to teach maintaining a cache and also fails to teach loading help data for the referenced component into the cache if the help data for a referenced component is not in the cache. In contrast, Martinez explicitly teaches that if no help information is found in dynamic table 41, help text may be loaded from static table 40 and displayed to the user, but it is not loaded into any kind of cache. Nor is it loaded from static table 40 to dynamic table 41. Additionally, if no help information is found in static table 40, the help system does not load or display any help text for the referenced object but simply returns to monitoring the current screen location of the mouse pointer.

Applicant submits that, contrary to the Examiner’s assertion, neither Cook nor Martinez, alone or in combination, teach maintaining a cache specific to help data for one or more user interface components; and in response to receiving a request for help data for a newly referenced one of the components: if the help data for the referenced component is not in the cache, loading the help data for the referenced component into the cache. Furthermore, as explained above, any system resulting from the Examiner’s proposed combination of Cook and Martinez would not include maintaining a cache specific to help data for on or more user interface components, nor would such a proposed system include loading the help data for a referenced component into the cache if the help data for a referenced component is not in the cache. Thus, the rejection of

claim 21 is clearly not supported by the teachings of the cited art and withdrawal thereof is respectfully requested. Similar arguments apply in regard to claims 35, 50 and 64.

Furthermore, regarding claim 23, Martinez fails to teach deleting the least recently requested help data in the cache if there is not enough free space in the cache to store the help data for the referenced component. The Examiner's cited reference (Martinez, column 5, lines 53-54) refers to the ability of application aware of Martinez help system to update the dynamic help text displayed for user interface objects. The cited section does not, nor does any other location in Martinez, mention deleting the least recently requested help data if there is not enough free space in the cache to store the help data for the referenced component. Instead, Martinez describes how "[a]n ideal use" for updating dynamic table 41 "would be to reveal details about objects which change dynamically, such as the number and type of mail items contained in an in-basket icon" (Cook, column 5, lines 75-60). Thus, the Examiner's cited passage deals with an aware application dynamically changing the context of strings displayed as help text to the user, not with deleting the least recently requested help data in the cache if there is not enough free space in the cache to store the help data for the referenced component.

Neither Cook nor Martinez, separately or in combination, teach deleting the least recently requested help data in the cache if there is not enough free space in the cache to store the help data for the referenced component. Thus, the rejection of claim 23 is not supported by the teachings of the cited art and withdrawal thereof is respectfully requested. Similar arguments apply in regard to claims 37, 52 and 66.

Regarding claim 24, Martinez fails to teach loading into the cache help data for one or more non-referenced user interface components *associated with a referenced component*. The Examiner cites column 6, lines 7-9 of Martinez, but this passage refers to how Martinez' system works with applications that do not provide their own help data. Specifically, the cited passage describes how such an application is "simply ignored by Infomouse 35" (Martinez, column 6, lines 1-3). Martinez does state that his help system can provide help data for generic user interface components even when working with an

application that does not provide its own help data. However, neither this passage, nor the entirety of Martinez, teaches the loading of help data for one or more non-referenced user interface components *associated with a referenced component*.

Furthermore, as described above regarding claim 21, Cook fails to teach a cache specific to help data and, as the Examiner admits, Cook further fails to teach loading into the cache help data for one or more non-referenced user interface components associated with a referenced component. Additionally, the Examiner's proposed combination of Cook and Martinez would not result in a system that would include loading into the cache help data for one or more non-referenced user interface components associated with a referenced component. Thus, the rejection of claim 24 is not supported by the teachings of the cited art and withdrawal thereof is respectfully requested. Similar arguments apply in regard to claims 29, 38, 43, 53, 58 and 67.

Regarding claim 32, Martinez does not teach in response to receiving a preload request indicating one or more of the components, loading into the cache help data for each of the indicated components. In fact, Martinez fails to teach anything regarding receiving a preload request indicating one or more components. Nor does Martinez teach anything about loading, into a cache, help data for each of the indicated components in response to receiving such a preload request. The Examiner's cited passage (Martinez, column 5, lines 39-44 and FIG. 3) refers only to the fact that Martinez' preferred operating system is IBM's OS/2 and that his help facility 35 works with three different types of applications: aware applications capable of dynamically modifying displayed help data, applications that only display static help data, and applications that do not provide any help data. Nowhere, including in the cited passage, does Martinez mention anything about receiving a preload request indicating one or more components nor about in response to receiving a preload request indicating one or more of the components, loading into the cache help data for each of the indicated components.

In contrast, as described above regarding claim 21, Martinez teaches a help system that include two string tables, dynamic table 41 and static table 40, which contain

help data to be displayed when user interface elements are referenced. However, no additional help data is loaded into either of these tables in response to a preload request indicating user interface components. Further, as the Examiner admits, Cook also fails to disclose such a system. Thus, the proposed combination of Cook and Martinez would also fail to include receiving a preload request indicating one or more components and would also clearly fail to include in response to receiving a preload request indicating one or more of the components, loading into the cache help data for each of the indicated components.

Thus, the rejection of claim 32 is not supported by the teachings of the cited art and withdrawal thereof is respectfully requested. Similar arguments apply in regard to claims 47 and 61.

Claims 26-28, 40-42 and 55-57 are rejected under U.S.C. § 103(a) as being unpatentable over Cook in view of Martinez, and further in view of Medl et al. (U.S. Patent 6,209,006) (hereinafter “Medl”). Applicants traverse this rejection for at least the reasons given above in regard to the respective independent claims.

Applicants also assert that the rejections of numerous ones of the dependent claims are further unsupported by the cited art. However, since the rejections of each of the independent claims have been shown to be improper, a further discussion of the rejections of the dependent claims is not necessary at this time.

## CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Applicant hereby petitions for such extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-85700/RCK.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☐ Petition for Extension of Time
- ☐ Notice of Change of Address
- ☐ Fee Authorization Form authorizing a deposit account debit in the amount of \$  
for fees (        ).
- ☐ Other:

Respectfully submitted,



Robert C. Kowert  
Reg. No. 39,255  
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8850

Date: November 12, 2004